Rworksheet_cadiz#3a

John Dave R. Cadiz

2024-10-02

Using Vectors

1. There is a built-in vector LETTERS contains the uppercase letters of the alphabet and letters which contains the lowercase letters of the alphabet.

LETTERS

[1] “A” “B” “C” “D” “E” “F” “G” “H” “I” “J” “K” “L” “M” “N” “O” “P” “Q” “R” “S”

[20] “T” “U” “V” “W” “X” “Y” “Z”

letters

[1] “a” “b” “c” “d” “e” “f” “g” “h” “i” “j” “k” “l” “m” “n” “o” “p” “q” “r” “s”

[20] “t” “u” “v” “w” “x” “y” “z”

##Based on the above vector Letters:

```
## a. You need to produce a vector that contains the first 11 letters.
first11 <- LETTERS[1:11]
print(first11)
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K"
```

```
## b. Produce a vector that contains the odd numbered letters.
oddlet <- LETTERS[seq(1,length(LETTERS), by = 2)]
print(oddlet)
```

```
##  [1] "A" "C" "E" "G" "I" "K" "M" "O" "Q" "S" "U" "W" "Y"
```

```
## c. Produce a vector that contains the vowels.
v <- LETTERS[c(1, 5, 9, 15, 21)]
print(v)
```

```
## [1] "A" "E" "I" "O" "U"
```

Based on the above vector letters:

```
## d. Produce a vector that contains the last 5 lowercase letters.
last5 <- tail(letters, 5)
print(last5)
```

```
## [1] "v" "w" "x" "y" "z"
```

```r
## e. Produce a vector that contains letters between 15 to 24 letters in lowercase.
let15n24 <- letters[15:24]
print(let15n24)
```

```
##  [1] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
```

Create a vector(not a dataframe) with the average temperatures in April for Tuguegarao City, Manila, Iloilo City, Tacloban, Samal Island, and Davao City. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees.

```r
## a. What is the R code and its result for creating a character vector for the city/town of Tuguegarao
city <- c("Tuguegarao City", "Manila", "Iloilo City", "Tacloban", "Samal Island", "Davao City")
print(city)
```

```
## [1] "Tuguegarao City" "Manila"          "Iloilo City"     "Tacloban"
## [5] "Samal Island"    "Davao City"
```

```r
## b. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees. Name the object as te
temp <- c(42, 39, 34, 34, 30, 27)
print(temp)
```

```
## [1] 42 39 34 34 30 27
```

```r
## c. Create a dataframe to combine the city and the temp by using 'data.frame(). What is the R code an
cittem <- data.frame(City = city, Temperature = temp)
print(cittem)
```

```
##              City Temperature
## 1 Tuguegarao City          42
## 2          Manila          39
## 3     Iloilo City          34
## 4        Tacloban          34
## 5    Samal Island          30
## 6      Davao City          27
```

```r
## d. Associate the dataframe you have create in 2.(c) by naming the columns using the names() function
names(cittem) <- c("City", "Temperature")
print(cittem)
```

```
##              City Temperature
## 1 Tuguegarao City          42
## 2          Manila          39
## 3     Iloilo City          34
## 4        Tacloban          34
## 5    Samal Island          30
## 6      Davao City          27
```

```r
## e. Print the structure by using str() function. Describe the output.
str(cittem)
```

```
## 'data.frame':    6 obs. of  2 variables:
##  $ City       : chr  "Tuguegarao City" "Manila" "Iloilo City" "Tacloban" ...
##  $ Temperature: num  42 39 34 34 30 27
```

```r
## - The output of the str(cittem shows that the cittem is a dataframe with 6 observations and 2 variab
```

```r
## f. From the answer in d, what is the content row 3 and row 4. What is its R code and its output?
row3n4 <- cittem[3:4, ]
print(row3n4)
```

```
##          City Temperature
## 3 Iloilo City          34
## 4    Tacloban          34
```

```r
## g. From the answer in d, display the city with highest temperature and the city with the lowest temp
hight <- cittem[which.max(cittem$Temperature), ]
lowt <- cittem[which.min(cittem$Temperature), ]
print(hight)
```

```
##              City Temperature
## 1 Tuguegarao City          42
```

```r
print(lowt)
```

```
##         City Temperature
## 6 Davao City          27
```

Using Matrices

- Matrix can be created by specifying the rows and columns.

```r
# row = 2

matrix(c(5, 6, 7, 4, 3, 2, 1, 2, 3, 7, 8, 9), nrow = 2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    5    7    3    1    3    8
## [2,]    6    4    2    2    7    9
```

```r
# row = 3 and column = 2
matrix(data = c(3, 4, 5, 6, 7, 8), 3, 2)
```

```
##      [,1] [,2]
## [1,]    3    6
## [2,]    4    7
## [3,]    5    8
```

```r
# creating a diagonal matrix where x value will always be 1
diag(1, nrow = 6, ncol = 5)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [6,]    0    0    0    0    0
```

```r
diag(6)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
## [4,]    0    0    0    1    0    0
```

```
## [5,]    0    0    0    0    1    0
## [6,]    0    0    0    0    0    1
```

2. Create a matrix of one to eight and eleven to fourteen with four columns and three rows.

```r
## a. What will be the R code for the #2 question and its result?
mmatrix <- matrix(c(1:8, 11:14), nrow = 3, byrow = TRUE)
print(mmatrix)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]   11   12   13   14
```

```r
## b. Multiply the matrix by two. What is its R code and its result?
multipliedm <- mmatrix * 2
print(multipliedm)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    4    6    8
## [2,]   10   12   14   16
## [3,]   22   24   26   28
```

```r
## c. What is the content of row 2? What is its R code?
row2 <- mmatrix[2, ]
print(row2)
```

```
## [1] 5 6 7 8
```

```r
## d. What will be the R code if you want to display the column 3 and column 4in row 1 and row 2? What
col3n4row1n2 <- mmatrix[1:2, 3:4]
print(col3n4row1n2)
```

```
##      [,1] [,2]
## [1,]    3    4
## [2,]    7    8
```

```r
## e. Whar is the R code if you want to display only the columns in 2 and 3, row 3? What is its output?
col2n3row3 <- mmatrix[3, 2:3]
print(col2n3row3)
```

```
## [1] 12 13
```

```r
## f. What is the R code if you want to display only the columns 4? What is its R code and correspondin
col4 <- mmatrix[, 4]
col4
```

```
## [1]  4  8 14
```

```r
## g. Name the rows as isa, dalawa, tatlo, and columns as uno, dos, tres, quatro for the matrix that wa

mmatrix <- matrix(c(1:8, 11:14), nrow = 3, byrow = TRUE)
rownames(mmatrix) <- c("isa", "dalawa", "tatlo")
colnames(mmatrix) <- c("uno", "dows" ,"tres", "quatro")
print(mmatrix)
```

```
##        uno dows tres quatro
## isa      1    2    3      4
```

```
## dalawa   5    6    7      8
## tatlo   11   12   13     14
```

```
## h. From the original matrix you have created in a, reshape the matrix by assigning a new dimension w

dim(mmatrix) <- c(6,2)
print(mmatrix)
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    5    7
## [3,]   11   13
## [4,]    2    4
## [5,]    6    8
## [6,]   12   14
```

Using Arrays

- Array can have more than two dimensions by using the array() function and dim() to specify the dimensions

```
# create a two-dimensional array containing numbers from 1 to 24 that have 3 rows and 4 columns

array_dta <- array(c(1:24), c(3, 4, 2))
print(array_dta)
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]   13   16   19   22
## [2,]   14   17   20   23
## [3,]   15   18   21   24
```

```
# checking for the dimensions

# row, columns, dimensions
dim(array_dta)
```

```
## [1] 3 4 2
```

```
# checking for the number of elements
length(array_dta)
```

```
## [1] 24
```

- Another way to create arrays

```
vectorA <- c(1:24)
an_Array <- array(vectorA, dim = c(3,4,2))
print(an_Array)
```

3. An arrays contains 1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1

```r
## a. Create an array above numeric values. Each Values will be repeated twice. What will be the R code

numval <- c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1)
array3d <- array(rep(numval, each = 2), dim = c(2, 4, 3))
print(array3d)
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    6
## [2,]    1    2    3    6
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]    7    8    9    0
## [2,]    7    8    9    0
##
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,]    3    4    5    1
## [2,]    3    4    5    1
```

```r
## b. How many dimensions do your array have?
# - The array that i have has three dimentsions. which have 2 rows, 4 columns, and 3 layers.
```

```r
## c. Name the rows as lowercase letters and coulumns as uppercase letters starting from the A. The arr

rownames(array3d) <- letters[1:2]
colnames(array3d) <- LETTERS[1:4]
dimnames(array3d) <- list(1:2, LETTERS[1:4], c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-D
print(array3d)
```

```
## , , 1st-Dimensional Array
##
##   A B C D
## 1 1 2 3 6
## 2 1 2 3 6
##
## , , 2nd-Dimensional Array
##
##   A B C D
## 1 7 8 9 0
## 2 7 8 9 0
##
## , , 3rd-Dimensional Array
##
##   A B C D
## 1 3 4 5 1
## 2 3 4 5 1
```